IN THE SPECIFICATION

Please amend the first full paragraph beginning on page 11, line 10 to page 12, line 13, as follows:

One purpose of incremental statistical timing is to respond to timing queries after one or more changes is/are made to the electrical circuit being timed or the conditions under which the electrical circuit was timed. To be efficient, retiming the entire circuit is avoided, hence the term "incremental." So the second input to incremental statistical timing is one or more changes, box 110 ~~120~~, which may be addition of a gate, re-sizing of a gate, addition of a latch, removal of a latch, addition of a wire, removal of a wire, buffering of a wire, etc. Box 120 shows the third input to the system, which is one or more statistical timing queries. Timing queries can include a request for a statistical slack at a timing point, a statistical arrival time or required arrival time at a timing point, a statistical slew (rise/fall time) at a timing point and a request to list the most critical path or paths. The query may request the mean, variance, full distribution or certain confidence level of each of these statistical values. The query may also request the sensitivity of the statistical value to one or more global sources of variation, or ask for the random part of the statistical quantity. Queries are usually, but not always, specific to a certain node or portion of the electrical circuit. Incremental statistical timing is performed (box 130) to produce an answer to the query or queries (box 140). Typically, a calling program exploits incremental statistical timing analysis to automatically optimize or improve the electrical circuit, hence the incremental timer is often invoked a large number of times. Thus it is important for the incremental timer to be efficient and calculate as little information as possible to answer the query. The calling program often tries many combinations of alternatives in a trial-and-error attempt to improve the circuit; thus, after a query, the change or changes most recently applied are often undone. It is therefore all the more important to do as little work as possible in answering the query since many changes ultimately get undone. Thus incremental timing uses the concept of "lazy evaluation" to re-compute as little information as possible in response to a query, while carefully

2

Appl. No. 10/665,092

keeping track of which pieces of most-recently-computed timing information are valid and which are not.

3